**MySQL**

Install mysql

Install mysql workbench

Run this in terminal

export PATH=${PATH}:/usr/local/mysql/bin

**login**

mysql -u root -p

**Create database**

create database name;

create database products;

Query OK, 1 row affected (0.01 sec)

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| **products**  |

| sys |

+--------------------+

5 rows in set (0.00 sec)

**Delete database**

drop database name;

**Show database**

Show databases;

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

4 rows in set (0.02 sec)

**Open database**

use name;

mysql> use products;

Database changed

mysql> select database();

+------------+

| database() |

+------------+

| products |

+------------+

1 row in set (0.00 sec)

select database();

show tables;

**Create Table:**

create table amazon\_products (S\_No int, Product\_Name varchar(50),Price int);

**Show Tables**

mysql> show tables;

+--------------------+

| Tables\_in\_products |

+--------------------+

| amazon\_products |

+--------------------+

1 row in set (0.00 sec)

**open tables**

mysql> desc amazon\_products;

+--------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+-------------+------+-----+---------+-------+

| S\_No | int | YES | | NULL | |

| Product\_Name | varchar(50) | YES | | NULL | |

| Price | int | YES | | NULL | |

+--------------+-------------+------+-----+---------+-------+

3 rows in set (0.02 sec)

**ADD column**

mysql> alter table amazon\_products ADD offer int;

Query OK, 0 rows affected (0.10 sec)

Records: 0 Duplicates: 0 Warnings: 0

+--------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+-------------+------+-----+---------+-------+

| S\_No | int | YES | | NULL | |

| Product\_Name | varchar(50) | YES | | NULL | |

| Price | int | YES | | NULL | |

| offer | int | YES | | NULL | |

+--------------+-------------+------+-----+---------+-------+

4 rows in set (0.01 sec)

**Delete table**

mysql> drop table amazon\_products;

**again create table:**

mysql> create table Amazon\_products(

-> Product\_Name varchar(50) not null,

-> Product\_Brand varchar(50) not null,

-> Product\_Price int not null,

-> Offer int,

-> Product\_number int not null primary key auto\_increment);

Query OK, 0 rows affected (0.02 sec)

mysql> desc Amazon\_products

-> ;

+----------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+-------------+------+-----+---------+----------------+

| Product\_Name | varchar(50) | NO | | NULL | |

| Product\_Brand | varchar(50) | NO | | NULL | |

| Product\_Price | int | NO | | NULL | |

| Offer | int | YES | | NULL | |

| Product\_number | int | NO | PRI | NULL | auto\_increment |

+----------------+-------------+------+-----+---------+----------------+

5 rows in set (0.00 sec)

**CRUD**

Create

Read

Update

Delete

**Data insert into tables**

INSERT INTO Amazon\_products

->(Product\_Name,Product\_Brand,Product\_Price,Offer,Product\_number)

-> VALUES

-> ('i Phone 13 Pro Max','Apple',110000,0,111);

INSERT INTO Amazon\_products

-> (Product\_Name,Product\_Brand,Product\_Price,Offer,Product\_number)

-> VALUES

-> ('MacBook Pro','Apple',150000,2000,002),

-> ('AirPods2','Apple',17000,1200,003);

Query OK, 2 rows affected (0.01 sec)

Records: 2 Duplicates: 0 Warnings: 0

**Show Data**

mysql> SELECT \* FROM Amazon\_products;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| iPhone | Apple | 110000 | 1 | 1 |

| MacBook Pro | Apple | 150000 | 2000 | 2 |

| AirPods2 | Apple | 17000 | 1200 | 3 |

+--------------+---------------+---------------+-------+----------------+

3 rows in set (0.00 sec)

mysql> SELECT Product\_Name Product\_Price FROM Amazon\_products

-> ;

+---------------+

| Product\_Price |

+---------------+

| iPhone |

| MacBook Pro |

| AirPods2 |

+---------------+

3 rows in set (0.00 sec)

mysql> SELECT Product\_Name, Product\_Price FROM Amazon\_products;

+--------------+---------------+

| Product\_Name | Product\_Price |

+--------------+---------------+

| iPhone | 110000 |

| MacBook Pro | 150000 |

| AirPods2 | 17000 |

+--------------+---------------+

3 rows in set (0.00 sec)

**WHERE**

mysql> SELECT \* FROM Amazon\_products WHERE Product\_Price<50000;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| AirPods2 | Apple | 17000 | 1200 | 3 |

+--------------+---------------+---------------+-------+----------------+

1 row in set (0.00 sec)

**BETWEEN**

mysql> SELECT \* FROM Amazon\_products WHERE Product\_number BETWEEN 3 AND 5;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| AirPods2 | Apple | 17000 | 1200 | 3 |

| S21 | Samsung | 70000 | 2000 | 4 |

| Z series | Samsung | 150000 | 4000 | 5 |

+--------------+---------------+---------------+-------+----------------+

3 rows in set (0.01 sec)

**UPDATE**

mysql> UPDATE Amazon\_products

-> SET Offer =4000

-> WHERE Product\_Number=3;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT \* FROM Amazon\_products;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| iPhone | Apple | 110000 | 1 | 1 |

| MacBook Pro | Apple | 150000 | 2000 | 2 |

| AirPods2 | Apple | 17000 | 4000 | 3 |

| S21 | Samsung | 70000 | 2000 | 4 |

| Z series | Samsung | 150000 | 4000 | 5 |

+--------------+---------------+---------------+-------+----------------+

5 rows in set (0.00 sec)

**DELETE**

mysql> DELETE FROM Amazon\_products

-> WHERE Product\_Name='Z series';

Query OK, 1 row affected (0.01 sec)

mysql> SELECT \* FROM Amazon\_products;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| iPhone | Apple | 110000 | 1 | 1 |

| MacBook Pro | Apple | 150000 | 2000 | 2 |

| AirPods2 | Apple | 17000 | 4000 | 3 |

| S21 | Samsung | 70000 | 2000 | 4 |

+--------------+---------------+---------------+-------+----------------+

4 rows in set (0.00 sec)

**ORDER BY**

mysql> SELECT \* FROM Amazon\_products

-> ORDER BY Product\_Price;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| AirPods2 | Apple | 17000 | 4000 | 3 |

| S21 | Samsung | 70000 | 2000 | 4 |

| iPhone | Apple | 110000 | 1 | 1 |

| MacBook Pro | Apple | 150000 | 2000 | 2 |

+--------------+---------------+---------------+-------+----------------+

4 rows in set (0.00 sec)

By default, is ascending order

**For descending order**

mysql> SELECT \* FROM Amazon\_products

-> ORDER BY Product\_Price DESC;

+--------------+---------------+---------------+-------+----------------+

| Product\_Name | Product\_Brand | Product\_Price | Offer | Product\_number |

+--------------+---------------+---------------+-------+----------------+

| MacBook Pro | Apple | 150000 | 2000 | 2 |

| iPhone | Apple | 110000 | 1 | 1 |

| S21 | Samsung | 70000 | 2000 | 4 |

| AirPods2 | Apple | 17000 | 4000 | 3 |

+--------------+---------------+---------------+-------+----------------+

4 rows in set (0.00 sec)

**LIMIT**

SELECT \* FROM Amazon\_products LIMIT 5;

**Aggregation**

**MIN**

**MAX**

**MEAN**

**COUNT**

SELECT AVG(Offer) FROM Amazon\_products;

**LIKE**

SELECT \* FROM Amazon\_products

WHERE Product\_brand LIKE ‘A%’;

**JOIN**

Inner Join

Left Join

Right Join

Full Join

Compare Two tables of content

Let’s assume there are two tables

1.student

2.marks

Student table:

+------+--------+---------+-----------+

| S\_Id | S\_Name | S\_Class | S\_Address |

+------+--------+---------+-----------+

| 1 | Ramu | 8 | ggm |

| 2 | venu | 10 | pnm |

| 3 | janu | 7 | pnm |

| 4 | rani | 6 | ggm |

| 5 | roja | 8 | ggm |

| 6 | rajesh | 7 | pnm |

| 7 | vamsi | 9 | ggm |

| 8 | dhoni | 8 | pnm |

| 9 | kohli | 9 | pnm |

| 10 | raj | 10 | pnm |

+------+--------+---------+-----------+

**Marks Table:**

mysql> SELECT \* FROM marks;

+------+--------+---------+---------+-------+

| S\_Id | Telugu | English | Science | Total |

+------+--------+---------+---------+-------+

| 1 | 78 | 79 | 54 | 210 |

| 2 | 90 | 98 | 78 | 250 |

| 3 | 89 | 78 | 88 | 265 |

| 4 | 79 | 89 | 34 | 196 |

| 5 | 73 | 73 | 44 | 190 |

| 6 | 72 | 39 | 44 | 230 |

+------+--------+---------+---------+-------+

6 rows in set (0.00 sec)

**COMPARE TWO TABLES**

mysql> SELECT S\_Name,S\_Class,Telugu

-> FROM student,marks

-> WHERE student.S\_Id=marks.S\_Id;

+--------+---------+--------+

| S\_Name | S\_Class | Telugu |

+--------+---------+--------+

| Ramu | 8 | 78 |

| venu | 10 | 90 |

| janu | 7 | 89 |

| rani | 6 | 79 |

| roja | 8 | 73 |

| rajesh | 7 | 72 |

+--------+---------+--------+

6 rows in set (0.00 sec)

**Other Method using INNER JOIN**

mysql> SELECT S\_Name,S\_Class,Total

-> FROM student

-> INNER JOIN marks

-> ON student.S\_Id=marks.S\_Id;

+--------+---------+-------+

| S\_Name | S\_Class | Total |

+--------+---------+-------+

| Ramu | 8 | 210 |

| venu | 10 | 250 |

| janu | 7 | 265 |

| rani | 6 | 196 |

| roja | 8 | 190 |

| rajesh | 7 | 230 |

+--------+---------+-------+

6 rows in set (0.00 sec)

**LEFT JOIN**

mysql> SELECT S\_Name,Science

-> FROM student

-> LEFT JOIN marks

-> ON

-> student.S\_Id=marks.S\_Id;

+--------+---------+

| S\_Name | Science |

+--------+---------+

| Ramu | 54 |

| venu | 78 |

| janu | 88 |

| rani | 34 |

| roja | 44 |

| rajesh | 44 |

| vamsi | NULL |

| dhoni | NULL |

| kohli | NULL |

| raj | NULL |

+--------+---------+

**RIGHT JOIN**

mysql> SELECT S\_Name,Science

-> FROM student

-> RIGHT JOIN marks

-> ON

-> student.S\_Id=marks.S\_Id;

**FULL JOIN**

mysql> SELECT S\_Name,Science

-> FROM student

-> LEFT JOIN marks

-> ON

-> student.S\_Id=marks.S\_Id

UNION

SELECT S\_Name,Science

-> FROM student

-> RIGHT JOIN marks

-> ON

-> student.S\_Id=marks.S\_Id;